



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

# Primer Trabajo Práctico

14 de Abril de 2010

Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Bianchi, Mariano	92/08	bianchi-mariano@hotmail.com
Brusco, Pablo	527/08	pablo.brusco@gmail.com
Di Pietro, Carlos Augusto Lyon	126/08	cdipietro@dc.uba.ar



**Facultad de Ciencias Exactas y Naturales**

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## Índice

# Ejercicio 1

## Introducción

El primer problema del presente trabajo consistió en la implementación de un algoritmo capaz de dar solución a la ecuación

$$b^n \bmod (n) \quad (1)$$

haciendo uso de alguna de las técnicas algorítmicas aprendidas hasta el momento en la materia. Asimismo, la consigna dictaba que la complejidad final del algoritmo debería ser menor a  $O(n)$ . En pos de cumplimentar lo pedido se decidió usar la técnica de *Dividir & Conquistar*<sup>1</sup> para desarrollar el algoritmo. Esta técnica se caracteriza principalmente en dividir la instancia de un problema en instancias más pequeñas, atacar cada una de ellas por separado y resolverlas, para finalmente juntar sus resultados y así producir el resultado final.

## Detalles de implementación

La primera solución que se piensa casi de manera intuitiva es la de multiplicar  $n$  veces el número  $b$  y luego hallar el resto de dividir ese resultado por  $n$ .

$$\underbrace{b.b.b \dots b.b.b}_{n \text{ veces}} \bmod (n) = b^n \bmod (n) \quad (2)$$

Sin embargo, la complejidad ese algoritmo es  $O(n)$ , ya que se realizan  $n$  multiplicaciones y 1 división, razón por lo cual no cumple con lo pedido en la consigna.

No obstante, la idea anterior conduce a otra forma de encarar el problema. Veamos de qué se trata.

Sabemos que  $b^n \bmod (n)$  es el resto de dividir  $b^n$  por  $n$ . Llamemos  $x$  a ese resto. Luego:

$$x = b^n \bmod (n) , \quad \text{con } 0 \leq x < n \quad (3)$$

ya que el Algoritmo de División de Números Enteros asegura que el resto existe, es único y es un valor entre 0 y  $n$ .

Inmediatamente de lo anterior, por la definición de congruencia, se desprende que:

$$b^n \equiv x \pmod{n} \quad (4)$$

Luego, una de las propiedades de congruencias, nos asegura que:

$$a \equiv p(n) , \quad b \equiv q(n) \implies a.b \equiv p.q(n) , \quad \forall a, b, p, q, n \in \mathbb{Z} \quad (5)$$

por lo que, tomando en particular,  $a = b$  sigue que:

$$b \equiv p(n) \implies b.b \equiv p.p \Leftrightarrow b^2 \equiv p^2(n) \quad (6)$$

Sea ahora  $k$  el resto de la division de  $p^2$  por  $n$ ; luego:

$$p^2 \equiv k \pmod{n} , \quad \text{con } 0 \leq k < n \quad (7)$$

---

<sup>1</sup>Poner alguna referencia en donde se explique esta técnica